IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 July 2022, accepted 3 August 2022, date of publication 16 August 2022, date of current version 22 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3198954

METHODS

An Implementation of Real-Time Traffic Signs and Road Objects Detection Based on Mobile GPU Platforms

EMIN GÜNEY[1], CÜNEYT BAYILMIŞ[2], AND BATUHAN ÇAKAN[3]
[1]Department of Computer Engineering, Sakarya University of Applied Sciences, 54050 Sakarya, Turkey
[2]Department of Computer Engineering, Sakarya University, 54050 Sakarya, Turkey
[3]Elkon Marine and Electric Technologies Research and Development Center, Elkon Elektrik San. Tic. A.S., 34953 Istanbul, Turkey

Corresponding author: Emin Güney (eminguney@subu.edu.tr)

This work was supported by Sakarya University Scientific Research Foundation (Project No.: 2021-7-24-20).

ABSTRACT Thanks to the rapid development of computer vision and deep learning technologies, advanced driver assistance systems (ADAS) have recently become widespread. These systems aim to increase driving safety and reduce the number of traffic accidents. Modern cars usually have ADAS systems integrated into their electronics, but other vehicles do not have such an integrated system. This paper presents a portable and image-based ADAS system for real-time detection of traffic signs, vehicles, and pedestrians. To realize real-time detection, the developed system uses the YOLO v5 algorithm. This single-stage detector is very popular as it has high detection speed and accuracy. The model was trained on the Tesla P100 graphics processing unit (GPU) with nearly 2500 images and 8 hours using GTSRB and study-specific dataset to analyze the developed system. Then, the implementation metrics (F1 score, P, R, PR curves) were calculated to evaluate the training and testing performances of the model. In addition, the model was compared in low-power, high-performance embedded platforms and in a computer to measure the real-time performance. The comparison results for (Jetson Xavier AGX, Jetson Xavier Nx, Jetson Nano and Test PC) achieved a speed of (43.59, 23.17, 6.4 and 3.41) frames per second for real-time detection, respectively. Considering the excellent accuracy and high speed, this study will guide researchers in demonstrating the efficiency and suitability of real-time road object detection with YOLO v5 on mobile platforms.

INDEX TERMS Traffic sign detection and recognition (TSDR), advanced driver assistance systems (ADAS), YOLO v5, pedestrian and object detection, GPU mobile platforms, deep learning.

I. INTRODUCTION

With the growing population, the number of vehicles participating in traffic is increasing significantly day by day. This inevitably increases the risk of traffic accidents and fatalities due to various reasons such as drowsiness, fatigue, and road conditions. Many studies and various solutions have been developed to reduce these accident risks and improve driving safety from the past to the present. The technologies developed in this context generally form the infrastructure of ADAS and autonomous driving systems. ADAS are systems that help drivers and vehicles detect dangerous traffic situations and respond to them accurately and quickly [1]. These systems have become an important field of study with the development of technologies to enhance safety and comfort in the automotive sector [2]. Although different technologies are used to develop these systems [3], [4], camera-based solutions offer significant cost advantages. Other advantages include processing and analyzing images using rapidly developing computer vision technologies [5], [6]. Outside the vehicle, external environment sensing includes collecting information about the driving environment [7]. The main external environmental information includes nearby cars, pedestrians, traffic signs, traffic lights, and some objects [8]. Since the perceptual quality of environmental information is affected by weather, road, and lighting conditions, these factors should be taken into account when designing the system [9]. These systems aim to improve driver awareness

The associate editor coordinating the review of this manuscript and approving it for publication was Hassan Omar.

by monitoring driver behavior [10], [11]. Monitoring driver behavior requires the processing and analyzing data retrieved from driving [12]. Therefore, it is vital to develop ADAS systems in traffic [13], [14], [15]. On the other hand, the main challenge for these systems is that the detected objects can provide real-time information to the driver and the system should be implemented on the hardware with satisfactory accuracy and high speed [16], [17], [18], [19]. Generally, most of the studies used only a computer as the working environment rather than a mobile platform, which contributes to the portability and mobility of the system. However, the use of mobile GPU platforms is advantageous in terms of performance compared to price. Therefore, in this work, an image-based system for detecting various vehicles, pedestrians, and traffic signs is developed on a GPU and tested on embedded platforms to achieve real-time detection.

This study can be summarized with four items: (i) a deep learning-based system was implemented to detect traffic signs, vehicles and pedestrians in real-time; (ii) the proposed detection system was developed with the YOLO v5 model, one of the most state-of-the-art deep learning detection architectures; (iii) the dataset was expanded by collecting real images specific to the study, and the model was trained with these datasets for hours on the GPU; and (iv) the system developed to demonstrate the portability and mobility of the system has been tested on three mobile platforms and one test computer.

The study has several significant advantages. Firstly, the detection processes based on camera and computer vision reduced the cost. Another advantage is that testing our real-time system on embedded platforms has contributed to efficiency and mobility. In addition, a recent comparison of a state-of-the-art detection algorithm (YOLO v5) on trendy embedded platforms has made the study up to date. In this way, this study is aimed to be a guide for researchers.

A. MOTIVATION

The main causes of traffic accidents are tired, drowsiness or drunk drivers getting involved in traffic and failing to notice the driving environment [20], [21], [22], [23]. A system to monitor and detect various vehicles and driver activities in traffic can help reduce the risk of accidents [24]. While the upper segments of cars are embedded in such ADAS systems that increase drivers' traffic awareness and improve the driving comfort [25], other vehicles do not have such a system embedded in the car [18]. This situation poses a high potential danger as it increases the safety risk of drivers on the road.

The motivation of this article is to implement an embedded detection application that can be integrated into the vehicle later in order to reduce the mentioned danger potential and guide those working in this field. Therefore, this paper also carried out various detection studies (traffic signs, road objects and pedestrian detection) for ADAS and autonomous driving on embedded platforms and the computer. For this, we trained the model on the study-specific dataset and the GTSRB dataset. Then, we tested this model on different platforms (Jetson Nano, Xavier Nx, Xavier AGX and test PC) for out-of-vehicle detections. Thus, we aimed to measure the efficiency of the system by running the model on these platforms.

B. CONTRIBUTIONS

The contributions of this paper are as follows:

- Deploy and test a state-of-the-art real-time deep learning architecture with high detection accuracy embedded systems and computer environment.
- Build an ADAS prototype for the low segment and autonomous vehicles that can later be integrated into the vehicle.
- Propose and implement a real-time ADAS system with low power consumption and high-speed mobile GPU platforms.
- Investigate and compare the efficiency and usability of mobile GPU platforms in order to detect and recognize real-time traffic signs and objects.
- Present a broad perspective with detection and analysis applications on this topic to ADAS researchers and developers.

The rest of this paper is organized as follows: Sect. 2 presents the related work of the study. Sect. 3 presents the experimental setup and detailed results of the analysis for the developed system. The obtained experimental results in numerous test environments are described in Sect. 4. Finally, Sect. 5 concludes the realized study and points out some possible future research directions.

II. RELATED WORKS

This section presents the detection and recognition of vehicles, pedestrians, and traffic signs in literature studies with various techniques. Many studies have been carried out recently for the traffic sign detection and recognition (TSDR) task in the driving environment. In two-stage detectors, the detection of traffic signs is performed in two stages: Detection and Recognition [18]. While the detection of the targeted object aims to focus on the detected object's location [26], a predictive classification process is applied in the identification process to determine the class to which the detected targets belong.

In general, TSDR studies can be classified as (i) vision-based [27], [28] and (ii) convolution-based approaches [29], [30], [31] together with traditional approaches [32]. Visual studies provide perception using the shape, edge, color, and light properties of the traffic signs [33]. Timofte *et al.* [34] combined 2D and 3D analysis to propose a multi-view scheme that considers color and shape-based criteria for traffic sign detection and recognition. The color-based system proposed by Li *et al.* [35] is essential in that it can work in different weather and illumination conditions. The three-stage approaches for TSDR studies by Yin *et al.* [36] helped to improve recognition accuracy and processing speed. Qian *et al.* [29] proposed a system based on deep

TABLE 1. Summary of related works.

References	Analysis Type	Application	Models	Techniques	Datasets
Li *et al.* 2015 [35]	Video frames	Traffic Sign Detection	Color segmentation, Shape symmetry based	Pyramid Histogram Oriented Gradients (PHOG)	Study-specific dataset
Yin *et al.* 2015 [36]	Video frames	Traffic Sign Recognition	Feature based Rotation Invariant Binary Pattern	Hough-SIFT transforms, Artificial Neural Networks (ANN)	GTSRB and STS
Qian *et al.* 2015 [29]	Video frames	Traffic Sign Detection	Based on multi-task CNN	Region proposal, edge detection and CCA Analysis	GTSRB, MNIST, CASIA GB1
Changzhen *et al.* 2016 [37]	Video frames	Traffic Sign Detection	Based on Faster R-CNN	Region Proposal Network (RPN)	Study-specific dataset
Berkaya *et al.* 2016 [38]	Images	Traffic Sign Detection and Recognition	Based on Edge Drawing Parameter Free (EDPF) algorithm	Local binary pattern, Gabor and HOG features	GTSRB dataset
Zhang *et al.* 2017 [39]	Real-time	Traffic Sign Detection and Recognition	Based on Enhanced YOLO v2	Grid partition technique and CNN network	GTSRB and CCTSDB
Ćorović *et al.* 2018 [40]	Real-time	Vehicle, Pedestrian and Traffic Sign Detection	Based on Darknet and YOLO v3	Grid partition technique and CNN network	Berkeley BDD100K
Xu *et al.* 2019 [41]	Video frames	Traffic Sign Detection	Based on Adaptive thresholding, Shape symmetry	Cumulative distribution function, Shape symmetry detection	GTSRB dataset
Balado *et al.* 2020 [42]	Images	Traffic Sign Detection and Recognition	Based on RetinaNet and Inception v3	Mobile mapping system, point clouds, data fusion	GTSRB dataset
Jin *et al.* 2020 [43]	Real-time	Traffic Sign Detection and Recognition	Based on Multi-Feature Single Shot Detector (SSD)	Feature fusion and enhancement techniques	GTSRB dataset
Wan *et al.* 2021 [44]	Images	Traffic Sign Detection and Recognition	Based on YOLO v3 architecture	Improved YOLO model and Grid partition technique	Tsinghua-Tencent 100K dataset
Liu *et al.* 2021 [32]	Images	Traffic Sign Detection and Recognition	Based on Cascade Saccade Network	Transferable visual features	TT100K, CCTSDB, and GTSRB
(Our system)	Real-time	Road Objects, Traffic Sign Detection and Recognition	Based on YOLO v5 architecture	Grid partition technique and CNN network	GTSRB and Study-specific dataset

convolutional neural network (DCNN) for TSDR, providing high detection and recognition accuracy performance.

Haloi [45] achieved high detection performances in six signal classes by training deep learning using the GTSRB dataset optimal parameters and memory. Changzhen *et al.* [37] presented a method based on DCNN using Region Proposal Network (RPN). Zhang *et al.* [39] detect traffic signs quickly and accurately in China, significantly reducing detection speed. YOLO v2-based systems have achieved very successful results in real-time detection. Ammour *et al.* [46] used Convolutional Neural Network (CNN) to detect cars using linear SVM classifier and mean shift algorithm. Ćorović *et al.* [40] introduced a YOLO v3-based system that performs robustly in real-time under various weather conditions. Xu *et al.* [41] proposed an innovative detection method for TSDR in a complex traffic environment using hypothesis testing for color threshold segmentation and shape symmetry. By creating a shape and color-based algorithm for data analysis, tests were carried out on the GTSRB dataset and high accuracy was obtained. Balado *et al.* [42] proposed a novel approach based on mapping traffic signs with the Mobile Mapping System for faster image processing. They made traffic sign detection with RetinaNet and classification with InceptionV3.

The related works are summarized comparatively in Table 1 according to the application metrics, analysis type, technology, method, and datasets. As can be seen from the table, the literature studies in this field generally have focused on detecting or recognizing traffic signs intensely, especially for safety and comfort. Within the scope of this paper, a YOLO v5-based system has been developed for real-time implementation of detection and recognition of vehicles, pedestrians and traffic signs. Then, the proposed system was

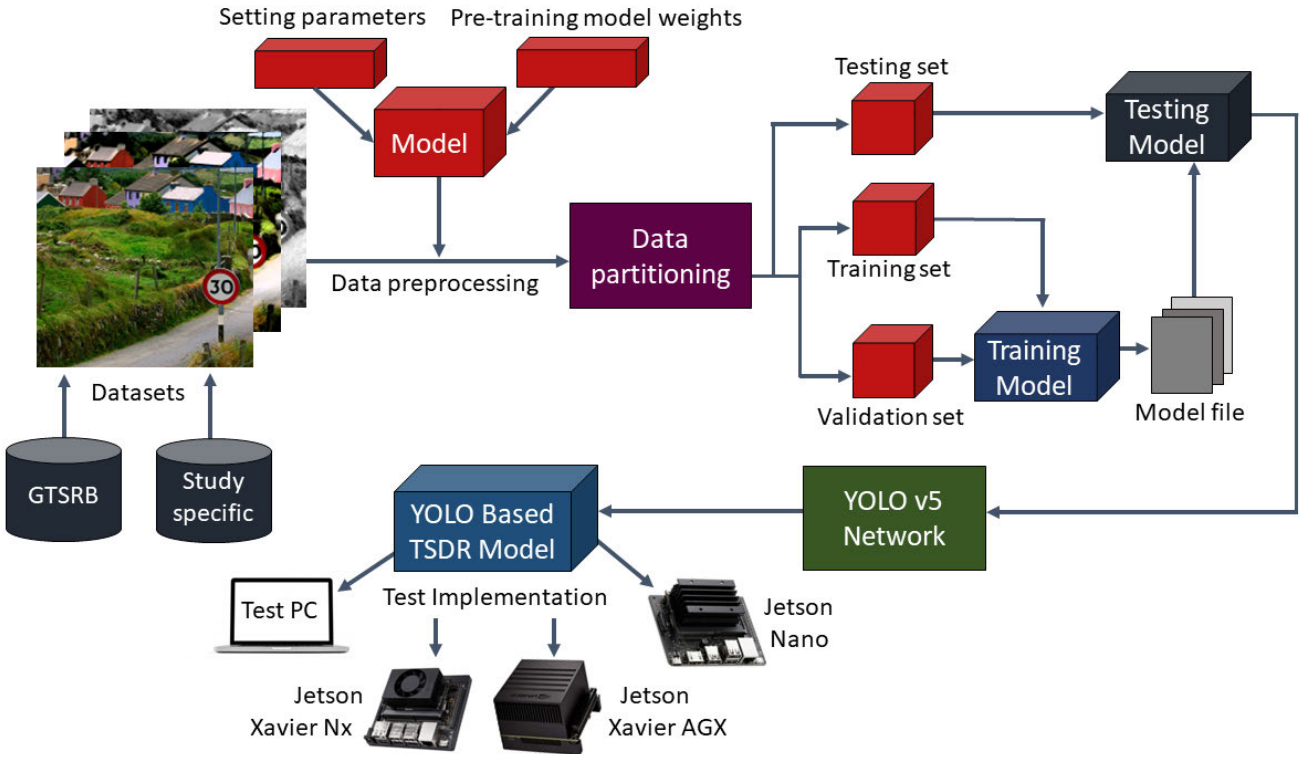


FIGURE 1. The proposed methodology for real-time traffic signs and road objects detection system.

compared to evaluate in low-power and high-performance embedded platforms.

III. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The system's architecture and the dataset used for this study are presented in this section. In addition, this section also describes the training and testing of the model in detail.

A. OVERVIEW OF THE PROPOSED ARCHITECTURE

This work follows the diagram in Figure 1 for real-time detection and recognition of traffic signs and road objects. The presented diagram for this work is based on a YOLO v5 model trained with Google Colab. The data collected in Turkey and the widely used benchmark for TSDR tasks German Traffic Sign Recognition Benchmark (GTSRB) were combined to develop the system considering the target classes. The dataset was organized by preprocessing the data. Using the YOLO v5 architecture, classes for development of the TSDR systems were trained with this dataset, and a trained model file is obtained with 80% training of our model and 20% in the testing process. This model was evaluated to measure the implementation metrics of training and testing performance in real-time. The evaluation of the proposed model was done by comparing the performance of the model tested on three different platforms.

1) DATASETS

The datasets of this work includes various class images from Turkey and the public traffic sign detection dataset GTSRB. The GTSRB dataset is a well-known dataset that contains various lighting and environmental conditions for evaluating TSDR performances. This dataset consists of 43 classes and includes traffic signs collected in Germany [47]. The study aims to detect 22 different traffic signs, pedestrians, and vehicles in Turkey by bringing together approximately 2500 images in total.

Details of the classes used are given in Figure 2. This figure showing the number of each data detected in the dataset provides information about the dataset. In addition, randomly selected images with different weather and light conditions collected from Turkey and the GTSRB datasets are demonstrated in Figure 3.

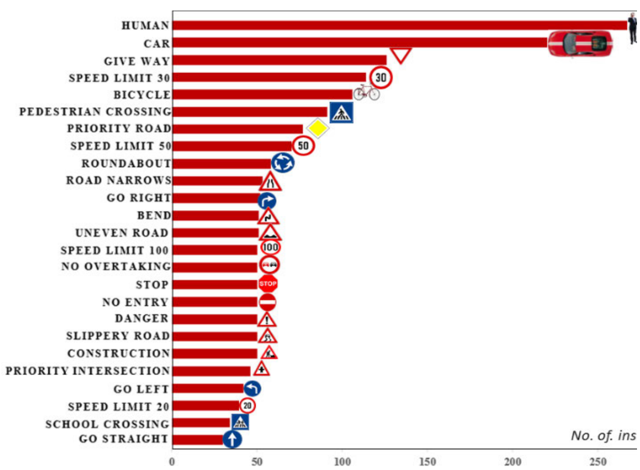


FIGURE 2. Summary of the dataset.

FIGURE 3. Sample images from datasets. The first row includes images collected from Turkey, and the second row includes the GTSRB dataset.

2) LABELLING PROCESS

All traffic signs and road objects are manually labeled separately to increase the classes' detection accuracy. The detection task is difficult due to the diversity and similarity of traffic signs. For example, in cases where an image contains more than one object and sign, as seen in Figure 4, the labeling process must be performed carefully. Moreover, the objects in the images were manually selected to determine the appropriate bounding boxes at this stage. Therefore, this stage has been a time-consuming and careful process.

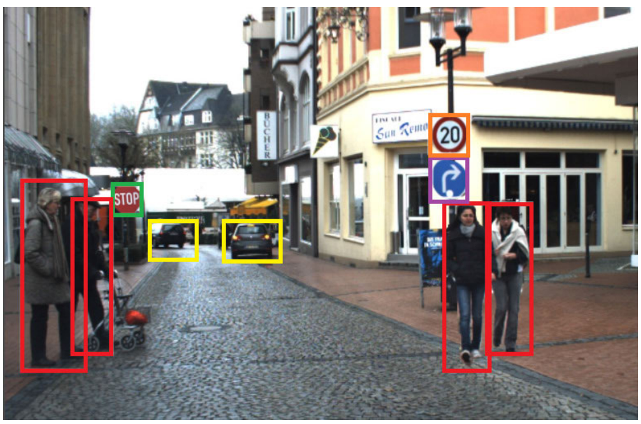


FIGURE 4. Labeling process for selecting target classes.

3) TRAINING STAGE

In order to obtain a robust system that works stably under different illumination and environmental conditions, the model must be trained with an appropriate dataset. At this stage, it is very important to determine in what proportion the dataset should be divided. In this work, data labeled for training in the dataset is divided between 80% and 20% for training and testing. The proposed system can detects vehicles, pedestrians, and traffic signs accurately and quickly with a camera in real-time. Deep learning applications require high computational power and processing speed due to too many hidden layers, constant weights updating, and increasing training parameters. To successfully perform the recognition operations, the model must be well trained. For this reason, the required model for the system proposed in this study was trained with the NVIDIA Tesla P100 GPU based on Pascal

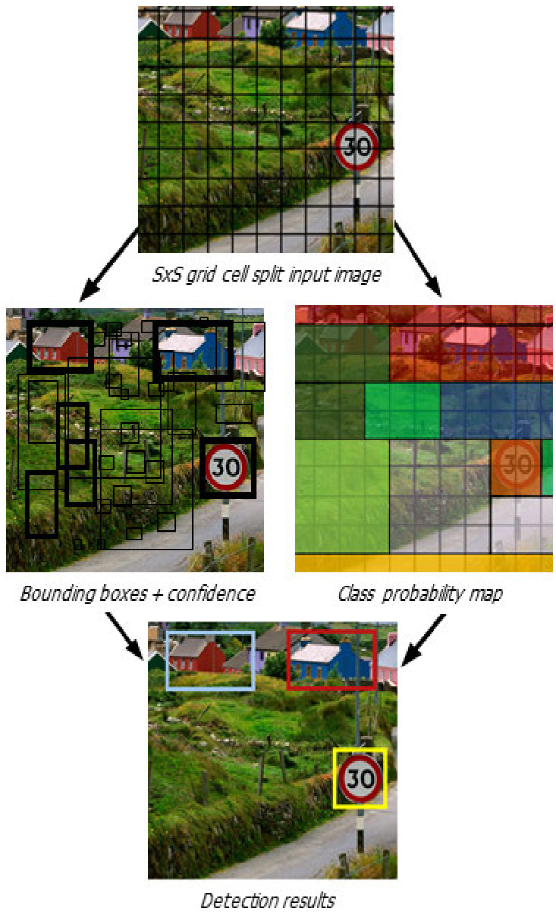


FIGURE 5. Conceptual design of the YOLO algorithm.

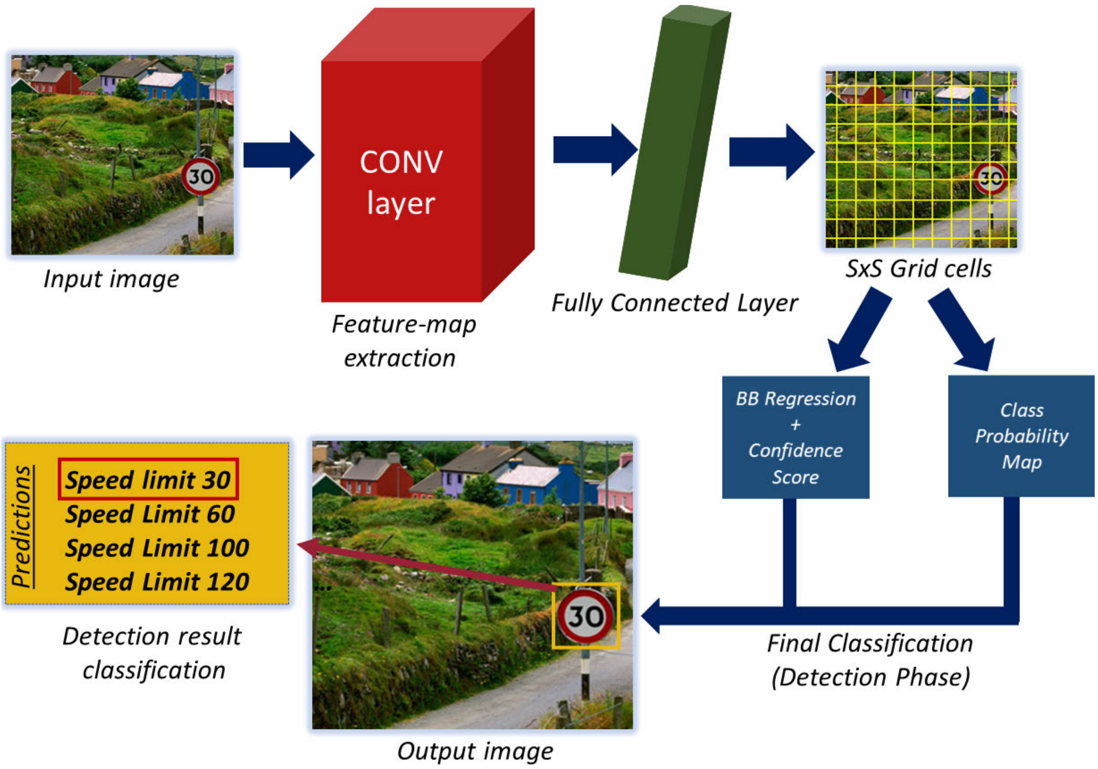


FIGURE 6. Object detection and classification process in YOLO.

architecture in Google Colab. The model was trained for about 8 hours with 850 epochs to detect vehicles, traffic signs, and pedestrians, and thus the model was prepared for testing.

B. DETECTION MODEL SELECTION
Object detection is the process of recognition and locating objects in images. Many architectures and models have been proposed in terms of accuracy and speed in the literature. This work used the YOLO v5 network to developed model.

1) YOLO NETWORK ARCHITECTURE
YOLO is a fast, high-performance real-time algorithm that uses CNN to better detect objects. Unlike previous algorithms, it performs the detection process with a regression-based approach [48]. Traditional object detection models, such as RCNN, offer a region of interest (RoI) for convolution [49], [50], while YOLO does detection and classification at one time. YOLO achieved this process by passing the input image through a single CNN network.

The input image is divided into regions and the bounding boxes and class probabilities are estimated for each region in YOLO. Its conceptual design repeats the process in real-time for each input image as shown in Figure 5. The algorithm's network model aims to detect cells that are responsible for detecting the object. The detection process of YOLO is shown in Figure 6. Each grid cell then assesses the B bounding boxes and their confidence scores for those boxes. These confidence scores show how confident the model is that the box contains an object and how accurately it predicted a box.

2) YOLO v5 BASED DETECTION AND RECOGNITION
YOLO v5 was released by Jocher *et al.* [51] shortly after YOLO v4. YOLO v5 has been compiled under a new training environment in PyTorch [52], [53]. This network model's detection accuracy and speed are relatively high compared to previous versions. Moreover, the size of the weighting file of the network model of YOLO v5 is about 90%.

The network's training has been carried out with YOLO v5s [54]. The YOLO v5 network architecture, the general structure given in Figure 7, consists of three parts: the spine, neck and head [55], [56]. Among these, CSPDarknet [57] is used on the spine, PANet [58] on the neck and YOLO

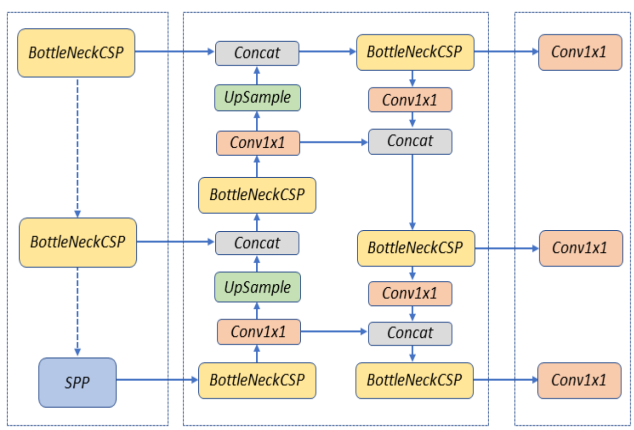


FIGURE 7. The YOLO v5 network structure diagram [55].

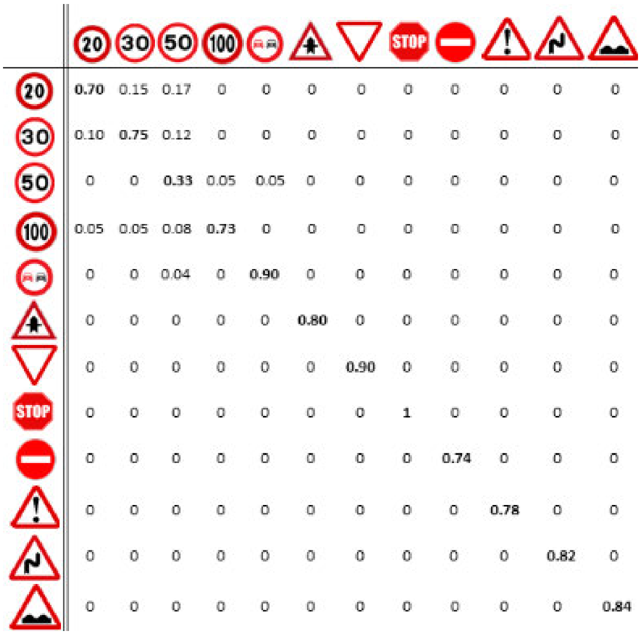


FIGURE 8. The first confusion matrix for the test results.

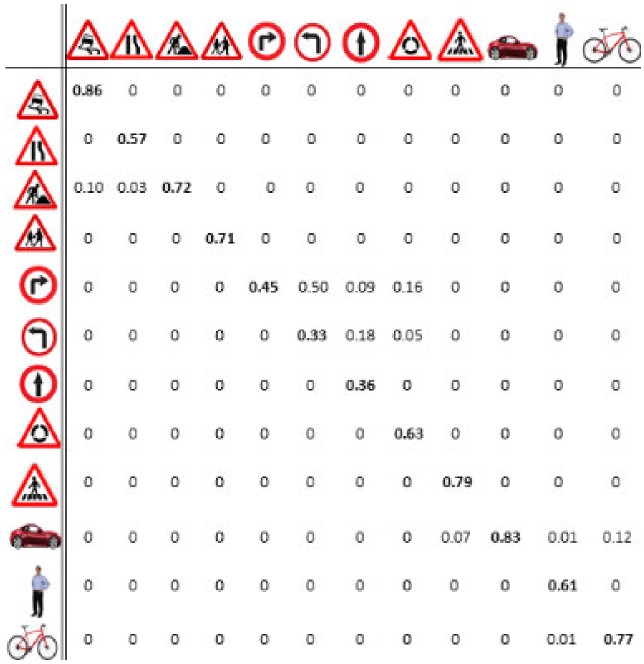


FIGURE 9. The other confusion matrix for the test results.

layer on the head. In summary, data is first passed through CSPDarknet for feature extraction and exported to PANet for feature aggregation. The YOLO layer gives outputs such as class, score, location and size information. CSPNet, which was used as a backbone in YOLO v4, was also used in YOLO v5 by updating some layers [54].

3) TESTING STAGE

This section presents the analysis results of the proposed model's effectiveness with various implementation metrics. The remaining 20% of the dataset was used to measure test performance. To reduce the dataset variance and prevent overfitting, the ideal ratios in the test model were chosen. The model's performance was analyzed using the confusion matrix shown in Figure 8.

The y-axis is the predicted labels, and the x-axis is the actual labels. The confusion matrix contains details about the values of the actual and predicted classes for classification and shows how accurately the model predicts the classes. The data in each row contains the model's estimated value for detection. For example, in Figure 8, the value of 0.75 highlighted in bold for the speed-30 traffic sign on line 3 indicates that the model predicted this sign with 75% accuracy. The 0.12 value marked on the right indicates that the 0.12 value was incorrectly estimated as the 50-speed limit. Detection errors due to background made estimation of the classes difficult. In addition, it can be seen that the determination is inconsistent for similar classes. Also, the confusion matrix is visualized to evaluate the performance similar to the other classes in Figure 9. The graph of the prediction accuracy of the model for each class can be easily seen in Figure 10. These accuracies are given for each class

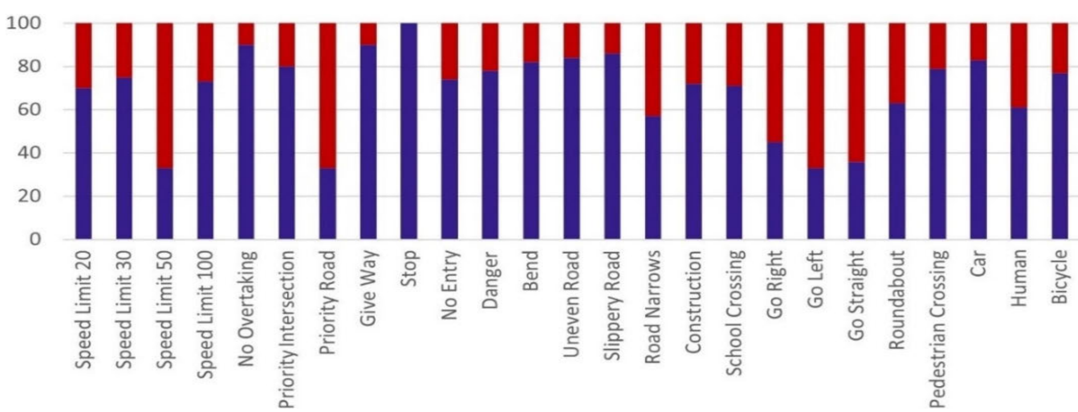


FIGURE 10. The accuracy graph of the proposed model for each class.

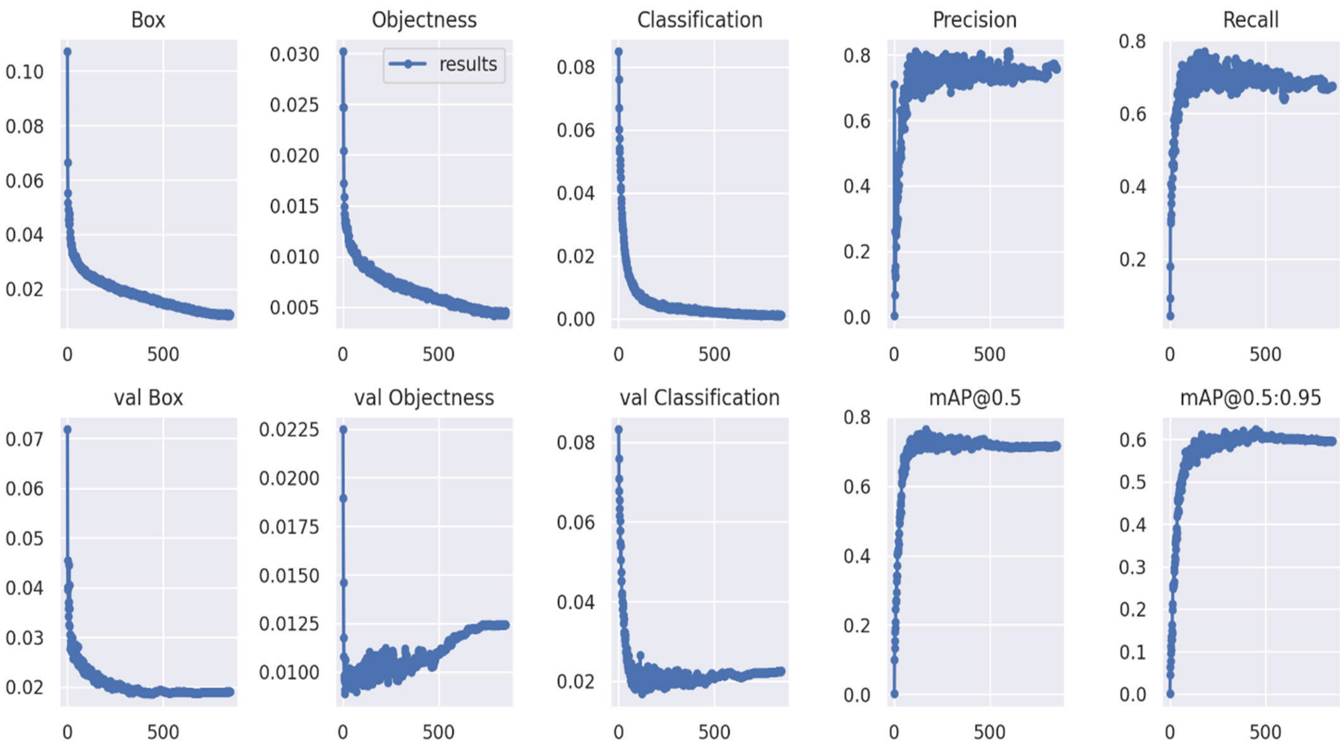


FIGURE 11. The performance metrics curves for the developed YOLO v5 model.

TABLE 2. The comparative results for training loss metric vs number of iterations with a step size.

Metrics	100	200	300	400	500	600	700	800	850
Objectness Loss	0.00662	0.00608	0.00453	0.00394	0.00266	0.00192	0.00106	0.00080	0.00058
Classification Loss	0.00898	0.00500	0.00416	0.00324	0.00240	0.00203	0.00138	0.00129	0.00129
Total Loss	0.02437	0.01932	0.01542	0.01254	0.00966	0.00728	0.00058	0.00389	0.00389

TABLE 3. The comparative results for validation loss metric vs number of iterations with a step.

Metrics	100	200	300	400	500	600	700	800	850
Objectness Loss	0.00427	0.00677	0.00515	0.00656	0.00780	0.01129	0.01273	0.01345	0.01357
Classification Loss	0.00893	0.00961	0.00798	0.00826	0.00771	0.00975	0.01110	0.01218	0.01286
Total Loss	0.01924	0.00948	0.00540	0.00285	0.00037	0.00307	0.00267	0.00291	0.00303

due to various tests also leading to differences in detection speed. The training results were automatically plotted in Figure 11 after the training was completed. Figure 11 signifies the developed model's performance with evaluation metrics, including Precision, Recall, and mAP (mean Average Precision).

The precise results of the training and testing are detailed given in Table 2 and also in Table 3. As seen from the tables, the different loss rates are compared through the various metrics, including objectness loss, classification loss, and total loss in the experiment. Over time, the decrease in loss value indicates a steady decline to about zero for

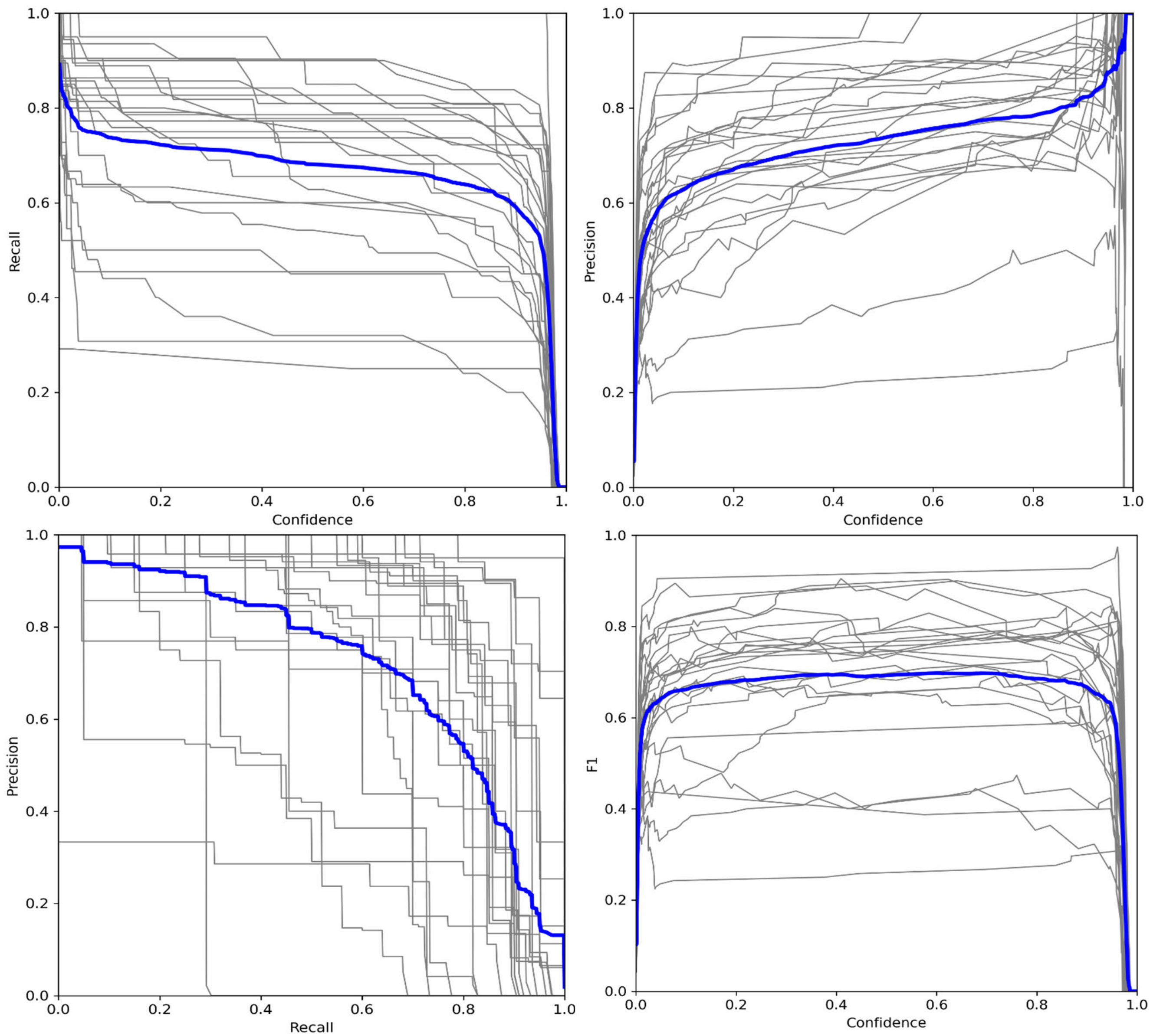


FIGURE 12. The model detection results of R-curve, P-curve, PR-curve and F1-curve for all classes.

TABLE 4. The average performance results in terms of various implementation metrics in Figure 12.

Model	P_Curve	PR_Curve	R_Curve	F1_Curve	Total training time
YOLO v5	0.986	0.716	0.89	0,70	7h+58min

the training loss metrics in Table 2. As can be understood, this decline is indicative of high performance improvement. The success rates for the validation step are similarly observed in Table 3. The values calculated for loss metrics are given in Table 2 and Table 3 according to the epochs. The implementation metrics were calculated for recall-confidence (R-curve), precision-confidence (P-curve), precision-recall (PR-curve), and F1-confidence (F1-curve) for each class in Figure 12.

At the same time, the average values of the trained model in Figure 12 were calculated, and the values in Table 4 were obtained. It is seen from Table 4 with the various application results that the model effectively detected the target classes. However it is necessary to draw comparison results in detail to

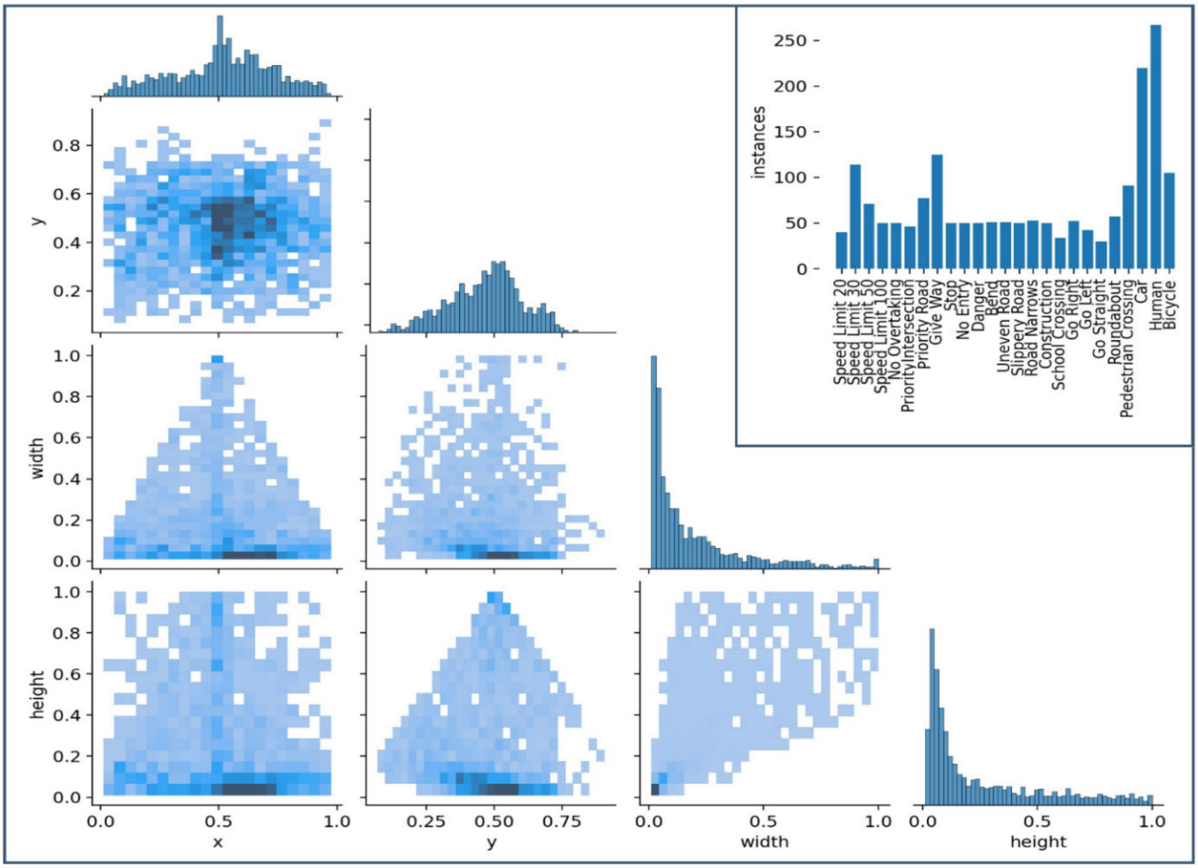


FIGURE 13. Labels correlogram for all classes.

understand the developed model's performance in detecting traffic signs and road objects. Therefore, labels' correlograms were analyzed in the study to visualize the statistical data analysis of the model effectively. The correlogram plot compares each data axis to every other axis in xywh space with detailed two-dimensional histogram groups. Figure 13 points to the label correlation plot to visualize the relationship of the bounding boxes. Figure 14 shows the YOLO v5 performs accurately detects traffic signs and road objects. From this figure, it can be seen from the detection rates that high performance is achieved.

IV. EXPERIMENTAL RESULTS

This section presents the real-time testing procedures of the developed model on mobile GPU platforms. To test the detection speed, the developed model was deployed and tested on Jetson Xavier AGX, Jetson Xavier Nx, Jetson Nano and test computer, respectively. The technical specifications for each platform are detailed in Table 5. From the table, it can be easily seen that the GPU platforms operate in real-time with a power consumption of 5 to 30 watts. In addition, the hardware of the platforms such as CPU, RAM and GPU are compared. These results are much lower than the power consumption of the test computer. Based on this, the proposed system performs much more efficiently and effectively in real-time on the embedded platforms.

TABLE 5. Test environments' technical specifications.

	Test PC	Xavier AGX	Xavier NX	Jetson Nano
CPU	Intel Core i7 6700 HQ	8-core Carmel ARM v8.2	6-core Carmel ARM v8.2	Quad-core ARM A57 MPCore
RAM	8 GB DDR4	32 GB 256-bit LPDDR	8 GB 128-bit LPDDR	4 GB 64-bit LPDDR
GPU	Geforce GTX 960 M 4 GB	512-core NVIDIA Volta™ GPU with 64 tensor	384-core NVIDIA Volta™ GPU with 48 tensor	128-core NVIDIA Maxwell™ GPU
AI performance	- - -	32 TOPs	21 TOPs	472 GFLOPs
Weight	2530 g.	280 g.	771 g.	241 g.
Camera	RaspBIN	TETRA	TETRA	SquaQ
Screen	15.6 inch	12.4 inch	12.4 inch	11 inch
Power cons.	120 Watt	30 Watt	15 Watt	5 Watt

Figure 15 graphically shows the real-time detection FPS rates by comparing four different platforms (Jetson Xavier AGX, Jetson Xavier Nx, Jetson Nano and test PC) to evaluate the developed system. The figure shows that Jetson Xavier AGX outperforms the other three platforms.

FIGURE 14. The test images and detection results with class indexes and confidence score.

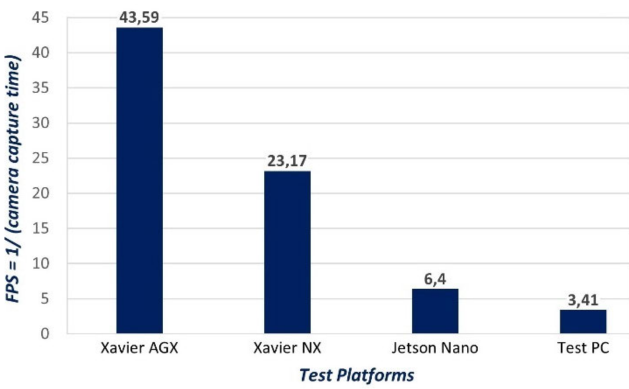


FIGURE 15. Comparison of real-time detection FPS speeds.

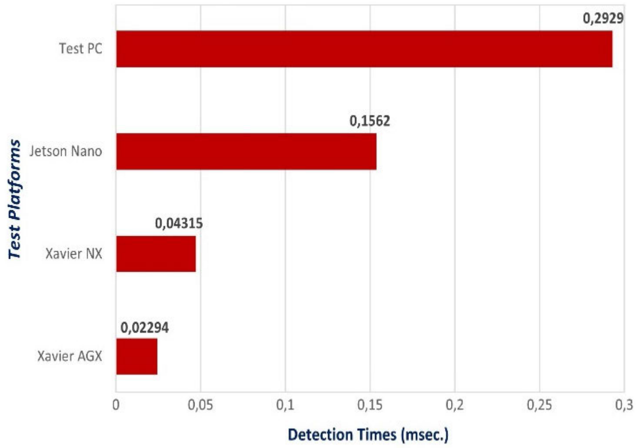


FIGURE 16. Comparison of real-time detection speed times.

In the study, real-time results of deep learning processes on Jetson Xavier AGX contributed significantly to maximum efficiency and performance. These values clearly indicate the proposed TSDR system contributes to speed and portability on state-of-the-art embedded platforms. As can be easily seen in the detection phase of the study, the highest speed was achieved with the NVIDIA Xavier AGX mobile platform. Figure 16 shows the detection times for real-time comparison of embedded platforms contributed significantly to speed and portability. As a result of the real-time tested work on embedded platforms, the use of YOLO v5 has resulted in high speed and accuracy, as can be seen.

Additionally, Jetpack Ubuntu 18.04 was installed on the embedded platforms used during testing. Moreover, many libraries such as PyTorch, CUDA, and CUDNN were installed. The developed system was successfully tested for robust and stable operation. Figure 17 shows an example of the real-time system implementation. Finally, the overall system has been successfully tested on the road to demonstrate the proposed real-time system. Our real-time system has great to efficiency and mobility.

FIGURE 17. Sample images from the real-time system implementation.

V. CONCLUSION AND FUTURE WORKS

This paper proposes a deep learning-based detection system for autonomous driving and driver assistance systems to contribute to mobility and portability. The proposed system implements applications on three different mobile GPU platforms that stand out in price and performance. For this purpose, the trend embedded platforms have implemented a real-time application that can detect vehicles, pedestrians, and traffic signs with the state-of-the-art YOLO v5 model. In the study, the model was developed with training and tests on the GPU and deployed on embedded platforms by real-time comparison. The system was developed using the GTSRB and the study-specific dataset with different lighting and weather conditions. We trained the model using the NVIDIA Tesla P100 GPU with about 8 hours of training for 850 epochs. Also, we discussed the factors affecting the performance of the training model. The trained model was tested on three different embedded platforms (Jetson Xavier AGX, Jetson Xavier Nx and Jetson Nano) and a test computer where the required configurations were made. The system's real-time performance was analyzed in detailed. The experimental results show that the Jetson Xavier AGX platform with low power consumption and high computational power provides high efficiency with the fastest inference speed and accuracy in detection.

Future studies plan to expand the dataset with various illumination and environmental conditions such as motion blur, color fade, undesirable light, occlusion, rain and snow. In addition, the proposed model will be developed with the latest technology deep learning models. It will contribute to the ADAS system to better detect road objects. Besides, it is aimed to investigate novel methods to increase detection speed and accuracy in real-time.

ACKNOWLEDGMENT

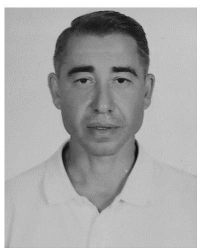
The authors would like to thank the anonymous reviewers for their useful suggestions.

REFERENCES

[1] M. Lu, K. Wevers, and R. Van Der Heijden, "Technical feasibility of advanced driver assistance systems (ADAS) for road traffic safety," *Transp. Planning Technol.*, vol. 28, no. 3, pp. 167–187, Jun. 2005.
[2] R. E. Haas, S. Bhattacharjee, and D. P. Möller, "Advanced driver assistance systems," in *Smart Technologies*. Singapore: Springer, 2020, pp. 345–371.
[3] W. Farhat, S. Sghaier, H. Faiedh, and C. Souani, "Design of efficient embedded system for road sign recognition," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 2, pp. 491–507, Feb. 2019.
[4] M. M. Iqbal, M. T. Mehmood, S. Jabbar, S. Khalid, A. Ahmad, and G. Jeon, "An enhanced framework for multimedia data: Green transmission and portrayal for smart traffic system," *Comput. Electr. Eng.*, vol. 67, pp. 291–308, Apr. 2018.
[5] L. Jian, Z. Li, X. Yang, W. Wu, A. Ahmad, and G. Jeon, "Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 81–86, May 2019.
[6] X. Qian, T. Lei, J. Xue, Z. Lei, and S. V. Ukkusuri, "Impact of transportation network companies on urban congestion: Evidence from large-scale trajectory data," *Sustain. Cities Soc.*, vol. 55, Apr. 2020, Art. no. 102053.
[7] Q. Li, W. Wu, L. Lu, Z. Li, A. Ahmad, and G. Jeon, "Infrared and visible images fusion by using sparse representation and guided filter," *J. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 254–263, May 2020.
[8] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.
[9] Y. Saadna and A. Behloul, "An overview of traffic sign detection and classification methods," *Int. J. Multimedia Inf. Retr.*, vol. 6, no. 3, pp. 193–210, Sep. 2017.
[10] A. Zeroual, F. Harrou, Y. Sun, and N. Messai, "Monitoring road traffic congestion using a macroscopic traffic model and a statistical monitoring scheme," *Sustain. Cities Soc.*, vol. 35, pp. 494–510, Nov. 2017.
[11] A. A. Minhas, S. Jabbar, M. Farhan, and M. Najam ul Islam, "Smart methodology for safe life on roads with active drivers based on real-time risk and behavioral monitoring," *J. Ambient Intell. Hum. Comput.*, pp. 1–13, Nov. 2019.
[12] C. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, and D. J. Ramsey, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, 2006.
[13] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition," in *Proc. 16th Mini-EURO Conf. 10th Meeting (EWGT)*, 2005, pp. 644–653.
[14] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-sign detection and tracking," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1329–1341, Sep. 2003.
[15] M. C. Kus, M. Gokmen, and S. Etaner-Uyar, "Traffic sign recognition using scale invariant feature transform and color classification," in *Proc. 23rd Int. Symp. Comput. Inf. Sci.*, Oct. 2008, pp. 1–6.
[16] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools Appl.*, vol. 79, pp. 18201–18219, Mar. 2020.
[17] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Proc. Eng.*, vol. 201, pp. 718–725, Dec. 2017.
[18] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.
[19] X. Bangquan and W. X. Xiong, "Real-time embedded traffic sign recognition using efficient convolutional neural network," *IEEE Access*, vol. 7, pp. 53330–53346, 2019.
[20] E. Petridou and M. Moustaki, "Human factors in the causation of road traffic crashes," *Eur. J. Epidemiol.*, vol. 16, no. 9, pp. 819–826, Sep. 2000.
[21] S. Park, F. Pan, S. Kang, and C. D. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 154–164.
[22] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57120–57128, 2019.
[23] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, "A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection," *IEEE Access*, vol. 8, pp. 29742–29754, 2020.

[24] W. J. R. Louwerse and S. P. Hoogendoom, "ADAS safety impacts on rural and urban highways," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 887–890.
[25] G. Velez and O. Otaegui, "Embedding vision-based advanced driver assistance systems: A survey," *IET Intell. Transp. Syst.*, vol. 11, no. 3, pp. 103–112, Apr. 2017.
[26] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gómez-Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.
[27] A. Madani and R. Yusof, "Traffic sign recognition based on color, shape, and pictogram classification using support vector machines," *Neural Comput. Appl.*, vol. 30, no. 9, pp. 2807–2817, Nov. 2018.
[28] J. M. Lillo-Castellano, I. Mora-Jiménez, C. Figuera-Pozuelo, and J. L. Rojo-Álvarez, "Traffic sign segmentation and classification using statistical learning methods," *Neurocomputing*, vol. 153, pp. 286–299, Apr. 2015.
[29] R. Qian, B. Zhang, Y. Yue, Z. Wang, and F. Coenen, "Robust Chinese traffic sign detection and recognition with deep convolutional neural network," in *Proc. Int. Conf. Natural Comput.*, Aug. 2015, pp. 791–796.
[30] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649.
[31] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2809–2813.
[32] Z. Liu, M. Qi, C. Shen, Y. Fang, and X. Zhao, "Cascade saccade machine learning network with hierarchical classes for traffic sign detection," *Sustain. Cities Soc.*, vol. 67, Apr. 2021, Art. no. 102700.
[33] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, Jul. 2016.
[34] R. Timofte, K. Zimmermann, and L. van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 633–647, 2014.
[35] H. Li, F. Sun, L. Wang, and L. Liu, "A novel traffic sign detection method via color segmentation and robust shape matching," *Neurocomputing*, vol. 169, no. 2, pp. 77–88, Dec. 2015.
[36] S. Yin, P. Ouyang, L. Liu, Y. Guo, and S. Wei, "Fast traffic sign recognition with a rotation invariant binary pattern based feature," *Sensors*, vol. 15, no. 1, pp. 2161–2180, Jan. 2015.
[37] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2016, pp. 676–679.
[38] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal, "On circular traffic sign detection and recognition," *Expert Syst. Appl.*, vol. 48, pp. 67–75, Apr. 2016.
[39] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2," *Algorithms*, vol. 10, no. 4, 2017.
[40] A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, "The real-time detection of traffic participants using YOLO algorithm," in *Proc. 26th Telecommun. Forum (TELFOR)*, Nov. 2018, pp. 1–4.
[41] X. Xu, J. Jin, S. Zhang, L. Zhang, S. Pu, and Z. Chen, "Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry," *Future Gener. Comput. Syst.*, vol. 94, pp. 381–391, May 2019.
[42] J. Balado, E. González, P. Arias, and D. Castro, "Novel approach to automatic traffic sign inventory based on mobile mapping system data and deep learning," *Remote Sens.*, vol. 12, no. 3, p. 442, Feb. 2020.
[43] Y. Jin, Y. Fu, W. Wang, J. Guo, C. Ren, and X. Xiang, "Multi-feature fusion and enhancement single shot detector for traffic sign recognition," *IEEE Access*, vol. 8, pp. 38931–38940, 2020.
[44] J. Wan, W. Ding, H. Zhu, M. Xia, Z. Huang, L. Tian, Y. Zhu, and H. Wang, "An efficient small traffic sign detection method based on YOLOv3," *J. Signal Process. Syst.*, vol. 93, no. 8, pp. 899–911, Aug. 2021.
[45] M. Haloi, "Traffic sign classification using deep inception based convolutional networks," 2015, *arXiv:1511.02992*.
[46] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens.*, vol. 9, no. 4, p. 312, Mar. 2017.
[47] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[49] R. Girshick, "Fast R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2015, pp. 1440–1448, 2015.
[50] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[51] G. Jocher, K. Nishimura, T. Mineeva, and R. Vilariño. (2020). *YOLOv5 Code Repository*. [Online]. Available: https://github.com/ultralytics/yolov5
[52] Z. Wang, Y. Wu, L. Yang, A. Thirunavukarasu, C. Evison, and Y. Zhao, "Fast personal protective equipment detection for real construction sites using deep learning approaches," *Sensors*, vol. 21, no. 10, p. 3478, May 2021.
[53] S. Li, X. Gu, X. Xu, D. Xu, T. Zhang, Z. Liu, and Q. Dong, "Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm," *Construct. Building Mater.*, vol. 273, Mar. 2021, Art. no. 121949.
[54] G. Jocher *et al.*, "ultralytics/yolov5: v4.0—nn.SiLU() activations Weights & Biases logging PyTorch Hub integration," Jan. 2021, doi: 10.5281/zenodo.4418161.
[55] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021.
[56] Q. Zhu, H. Zheng, Y. Wang, Y. Cao, and S. Guo, "Study on the evaluation method of sound phase cloud maps based on an improved YOLOv4 algorithm," *Sensors*, vol. 20, no. 15, p. 4314, Aug. 2020.
[57] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 390–391.
[58] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

EMIN GÜNEY was born in Istanbul, in 1996. He received the Graduate degree from the Computer Engineering Department, Sakarya University, and the M.Sc. degree from the Institute of Computer Science, Sakarya University, in 2021. He has been working as a Research Assistant with the Sakarya University of Applied Science. His current interests include advanced driver assistance systems (ADAS), autonomous driving systems, image processing, object detection, and data analysis.



CÜNEYT BAYILMIŞ received the B.Sc. and M.Sc. degrees from Sakarya University, Sakarya, Turkey, in 2001 and 2003, respectively, and the Ph.D. degree from Kocaeli University, Turkey, in 2006. He worked as a Postdoctoral Researcher with the University of Maryland, Baltimore County, USA, between May 2009 and January 2010. He is currently working as a Professor with the Department of Computer Engineering, Sakarya University. His research interests include the Internet of Things, wireless sensor networks, and embedded systems applications.

BATUHAN ÇAKAN received the B.Sc. degree in electrical and electronics engineering from Sakarya University, Sakarya, Turkey, in 2021. He is currently pursuing the M.Sc. degree in electrical and electronics engineering with the Sakarya University of Applied Sciences, Sakarya. Since 2021, he has been working at the Elkon Marine and Electric Technologies Research and Development Center.